

EK113979164US

INCASTING FOR DOWNLOADING
FILES ON DISTRIBUTED NETWORKS

P. OSCAR BOYKIN

1 BACKGROUND OF THE INVENTION

2 The present invention relates to a data storage/access
3 in a client-server system which consists of a plurality of
4 hosts each of which may act as either a server or clients
5 and which are interconnected by a shared communication
6 channel.

7 Research and development have been achieved on a
8 server with a storage device for storing a number of files,
9 such as a movie. The server distributes these files upon a
10 demand from a client.

11 A video server system needs extension due to lack of
12 capacity of server computers, it has been solved by
13 replacing the old ones with a higher performance server
14 computer, or by increasing the number of server computers

1 so that a load of processing is distributed over a
2 plurality of server-computers. The latter way of extending
3 the system by increasing the number of server computers is
4 effective in terms of workload and cost. A video server as
5 such is introduced in "A Tiger of Microsoft, United States,
6 Video on Demand" in an extra volume of Nikkei
7 Electronics titled "Technology which underlies Information
8 Superhighway in the United States", pages 40, 41 published
9 in Oct. 24, 1994 by Nikkei BP.

10 A server system includes a network and server-
11 computers. The server-computers are connected to the
12 network and have a function as a video server, magnetic
13 disk unit which are connected to the server computers and
14 stores video programs, clients which are connected to the
15 network and demand the server computers to read out a video
16 program. Each server computer has a different plurality of
17 set of video programs such as a movie stored in the

1 magnetic disk units. A client therefore reads out a video
2 program via one of the server-computers which has a
3 magnetic disk units where a necessary video program is
4 stored. The server system in which each one of a plurality
5 of server-computers stores an independent set of video
6 programs. The server system is utilized efficiently when
7 each demand on a video program is distributed to different
8 server computers. However when a plurality of accesses
9 rush into a certain video program, a work load increases on
10 a server computer where this video program is stored,
11 namely a work load disparity will be caused among server
12 computers. Even if the other server computers remain idle,
13 the whole capacity of the system has reached to the utmost
14 level because of the overload on a capacity of a single
15 computer. This deteriorates the efficiency of the server
16 system.

17 U. S. Patent No. 5,630,007 teaches a client-server

1 system which includes a plurality of servers and a
2 plurality of storage devices. The storage devices
3 sequentially store data. The data is distributed in each
4 of the plurality of storage devices. Each server device is
5 connected to the plurality of storage devices for accessing
6 the data distributed and stored in each of the plurality of
7 storage devices. The client-server system improves
8 efficiency of each server by distributing loads to a
9 plurality of servers. The client-server system also
10 includes an administration apparatus. The administration
11 apparatus is connected to the plurality of servers for
12 administering the data sequentially stored in the
13 plurality of storage devices and the plurality of servers.
14 A client is connected to both the administration apparatus
15 and the plurality of servers. The client specifies a
16 server that is connected to a storage device where a head
17 block of the data is stored by inquiring to the

1 administration apparatus and accesses the data in the
2 plurality of servers according to the order of the data
3 storage sequence from the specified server. The client
4 makes an inquiry to the administration apparatus and
5 accesses the data in the plurality of servers in accordance
6 to the order of the data storage sequence from the
7 specified server.

8 U. S. Patent No. 5,905,847 teaches a client-server
9 system which improves efficiency of each server by
10 distributing loads to a plurality of servers having a
11 plurality of storage devices. The storage devices
12 sequentially store data. The data is distributed in each
13 of the plurality of storage devices. Each server is
14 connected to the plurality of storage devices for accessing
15 the data distributed and stored in each of the plurality of
16 storage devices. An administration apparatus is connected
17 to the plurality of servers for administrating the data

1 sequentially stored in the plurality of storage devices and
2 the plurality of servers. A client is connected to both
3 the administration apparatus and the plurality of servers.
4 The client specifies a server which is connected to a
5 storage device in which a head block of the data is stored
6 by making an inquiry to the administration apparatus and
7 accesses the data in the plurality of servers in accordance
8 to the order of the data storage sequence from the
9 specified server.

10 U. S. Patent No. 5,926,101 teaches a multi-hop
11 broadcast network of nodes which have a minimum of hardware
12 resources, such as memory and processing power. The
13 network is configured by gathering information concerning
14 which nodes can communicate with each other using flooding
15 with hop counts and parent routing protocols. A
16 partitioned spanning tree is created and node addresses are
17 assigned so that the address of a child node includes as

1 its most significant bits the address of its parent. This
2 allows the address of the node to be used to determine if
3 the node is to process or resend the packet so that the
4 node can make complete packet routing decisions using only
5 its own address.

6 U. S. Patent No. 6,108,703 teaches a network-
7 architecture which has a framework. The framework supports
8 hosting and content distribution on a truly global scale.
9 The framework allows a content provider to replicate and
10 serve its most popular content at an unlimited number of
11 points throughout the world. The framework includes a set
12 of servers operating in a distributed manner. The actual
13 content to be served is preferably supported on a set of
14 hosting servers (sometimes referred to as ghost servers).
15 This content includes HTML page objects that are served
16 from a content provider site. A base HTML document portion
17 of a Web page is served from the content provider's site

1 while one or more embedded objects for the page are served
2 from the hosting servers, preferably, those hosting servers
3 near the client machine. By serving the base HTML document
4 from the content provider's site, the content provider
5 maintains control over the content.

6 U. S. Patent No. 5,367,698 teaches a networked digital
7 data processing system which has two or more client devices
8 and a network. The network includes a set of inter-
9 connections for transferring information between the client
10 devices. At least one of the client devices has a local
11 data file storage element for locally storing and providing
12 access to digital data files arranged in one or more client
13 file systems. A migration file server includes a migration
14 storage element that stores data portions of files from the
15 client devices, a storage level detection element that
16 detects a storage utilization level in the storage element,
17 and a level-responsive transfer element that selectively

1 transfers data portions of files from the client device to
2 the storage element.

3 U. S. Patent No. 5,802,301 teaches a method for
4 improving load balancing in a file server. The method
5 includes the steps of determining the existence of an
6 overload condition on a storage device having a plurality
7 of retrieval streams, accessing at least one file thereon,
8 selecting a first retrieval stream reading a file,
9 replicating a portion of the file being read by the first
10 retrieval stream onto a second storage device and reading
11 the replicated portion of the file on the second storage
12 device with a retrieval stream capable of accessing the
13 replicated portion of the file. The method enables the
14 dynamic replication of data objects to respond to
15 fluctuating user demand. The method is particularly useful
16 in file servers such as multimedia servers delivering
17 continuously in real time large multimedia files such as

1 movies.

2 U. S. Patent No. 5,542,087 teaches a data processing
3 method which generate a correct memory address from a
4 character or digit string such as a record key value and
5 which is adapted for use in distributed or parallel
6 processing architectures such as computer networks,
7 multiprocessing systems, and the like. The data processing
8 method provides a plurality of client data processors and a
9 plurality of file servers. Each server includes at least a
10 respective one memory location or "bucket". The data
11 processing method includes the steps of generating a key
12 value by means of any one of the client data processors and
13 generating a first memory address from the key value. The
14 first address identifies a first memory location. The data
15 processing method also includes the steps of selecting
16 from the plurality of servers a server that includes the
17 first memory location, transmitting the key value from the

1 one client to the server that includes the first memory
2 location and determining whether the first address is the
3 correct address by means of the server. The data
4 processing method further provides that if the first
5 address is not the correct address then performing the
6 steps of generating a second memory address from the key
7 value by means of the server, the second address
8 identifying a second memory location, selecting from the
9 plurality of servers another server which includes the
10 second memory location, transmitting the key value from the
11 server that includes the first memory location to the other
12 server which includes the second memory location,
13 determining whether the second address is the correct
14 address by means of the other server and generating a third
15 memory address, which is the correct address, if neither
16 the first or second addresses is the correct address. The
17 data processing method provides fast storage and subsequent

1 searching and retrieval of data records in data processing
2 applications such as database applications.

3 Distributed storage and sharing of data and program
4 files has become an integral part of doing business over
5 the Internet and other distributed networks. Such a
6 distributed environment is characterized by the fact that
7 multiple copies of the same file reside over the network.

8 In peer-to-peer networking each user also doubles as a
9 server connected to the Internet. Service providers, such
10 as Napster, Gnutella and Freenet have emerged. This
11 emerging technology has the potential to revolutionize
12 Internet and E-Commerce, but several technological
13 challenges have to be overcome before it can be translated
14 into a robust product which hundreds of millions of
15 customers can reliably use.

16 The most frequent use of such a network is for
17 downloading purposes. A client looks up the content list,

1 and wants to download a particular file/content from the
2 network. The existing protocols for this process are
3 extremely simple and can be described in general as
4 follows. The client or a central server searches the list
5 of servers that contain the desired file, and picks one
6 such server (either randomly or according to some priority
7 list maintained by the central server) and establishes a
8 direct connection between the client requesting the down
9 load and the chosen server. This connection is maintained
10 until the entire file has been transferred. The exact
11 implementation might vary from one protocol to another;
12 however, the fact that only one server is picked for the
13 transfer of the entire requested file remains invariant.

14 The above-mentioned existing protocols suffer from
15 several serious drawbacks, as stated next. Since only one
16 server is picked for the transfer of the entire file (even
17 though there are potentially many servers with the same

content), the quality of service becomes totally dependent on the bandwidth and the reliability of the Internet access that the chosen server maintains during the transfer. This poses a serious problem, especially in the case of networks that primarily comprise of low-performance servers as is the case for Napster and other proposed peer-to-peer networks and the reliability and speed of the host computers cannot be guaranteed. The average available bandwidth could be as low as that of a 28.8K or a 56K modem. Moreover, the connection of the server to the Internet could be dropped in the middle of a download, necessitating another attempt from the beginning. For example, an average MP3 file is around 5 Mega-bytes in length, and it will take around 16-20 minutes to download it over a 56K modem!! If the connection is dropped at any time during this period, then one needs to attempt the download all over again. The issue of choosing the best

1 server among those that have a copy of the requested file
2 is not properly addressed, leading to a further loss in the
3 quality of the service. If the winner is picked randomly
4 then clearly it is not the best choice. Even if the winner
5 is picked based on a pre-sorted list, where servers are
6 ranked according to their average available bandwidth, the
7 resulting scheme would be far from optimal. In particular,
8 even if a server has a higher average bandwidth, since it
9 comprises only a part of the host computer and shares the
10 bandwidth with other competing tasks, the available
11 bandwidth for the download could be drastically low during
12 the time of the transfer. The protocols do not take
13 advantage of the fact that the client could have a much
14 higher available bandwidth than any of the potential
15 servers. For example, even if the client is connected to a
16 high-speed Ethernet, the effective transfer rate for the
17 session could still be as low as that of a modem that the

1 chosen server might be using. Accuracy and integrity of
2 the downloaded file are not usually guaranteed. Since
3 multiple copies of the files are maintained by different
4 servers the issue of the integrity of the downloaded files
5 becomes a serious concern.

6 The inventor incorporates the teachings of the above-
7 cited patents into this specification.

8 SUMMARY OF THE INVENTION

9 The present invention is generally directed to a
10 distributed network which includes a plurality of hosts and
11 a shared communication channel. Each hosts is coupled to
12 the shared communication channel. Each host acts as both a
13 client and a server.

14 In a first separate aspect of the present invention,
15 the distributed network is used to incast fragments from
16 multiple copies of a file in order to be gathered together
17 so that a single copy of the file can be generated.

1 In a second separate aspect of the present invention,
2 at least one host has a global list with entries. Each
3 entry contains all the necessary information about a file.

4 The features of the present invention which are
5 believed to be novel are set forth with particularity in
6 the appended claims.

7 DESCRIPTION OF THE DRAWINGS

8 Fig. 1 is a schematic diagram of a video server system
9 of the prior art.

10 Fig. 2 is a schematic diagram of a video server system
11 of U. S. Patent No. 5,630,007.

12 Fig. 3 is a schematic diagram of an administration
13 table according to U. S. Patent No. 5,630,007.

14 Fig. 4 is a schematic drawing a distributed network
15 which has a plurality of hosts according to the present
16 invention wherein each host acts as both a client and a
17 server.

1 Fig. 5 is a schematic drawing of a file format for use
2 in the distributed network of Fig. 4.

3 Fig. 6 is a schematic drawing of an entry for a file
4 in a global list in which the entry contains all the
5 necessary information about the file so that a client can
6 successfully complete an incasting process using the
7 distributed network of Fig. 4.

8 DESCRIPTION OF THE PREFERRED EMBODIMENT

9 Fig. 1 is a video server system of the prior art
10 includes a network 1 and server computers 2. The server
11 computers 2 are connected to the network 1 and have a
12 function as a video server, magnetic disk unit 3 which are
13 connected to the server computers 2 and stores video
14 programs, clients 5 which are connected to the network 1
15 and demand the server computers 2 to read out a video
16 program. Each server computer 2 has a different plurality
17 of set of video programs such as a movie stored in the

1 magnetic disk units 3. A client 5 therefore reads out a
2 video program via one of the server computers 2 which has a
3 magnetic disk units 3 where a necessary video program is
4 stored.

5 Referring to Fig. 2 in conjunction with Fig. 3 a video
6 server system 10 of U. S. Patent No. 5,630,007 includes a
7 network 11, such as Ethernet and ATM, and a plurality of
8 server computers 12. Application programs are connected to
9 the network 11. Magnetic disk units 31 and 32 are
10 connected to the server computers which sequentially store
11 distributed data, such as a video program, which has been
12 divided (referred to as "striping") to be stored in the
13 magnetic disk units 31 and 32, client computers 5 which are
14 connected to the network 1 and receive video program,
15 application programs which operate in the client computers
16 5, driver programs as an access demand means which demand
17 access to the video program 4 having been divided and

1 sequentially stored in magnetic disk units 31 and 32 in
2 response to a demand to access from application programs.
3 Client-side network interfaces carry out such process as
4 TCP/IP protocol in the client computers 5 and realize
5 interfaces between clients and the network 1, server-side
6 network interfaces which carry out such processes as TCP/IP
7 protocol in the server computers 2 and realizes interface
8 between servers and the network 1, server programs which
9 read data block out of magnetic disk units 31 and 32 to
10 supply it to the server-side network interfaces the
11 original video program 11 which has not yet been divided
12 nor stored, administration computer 12 connected to the
13 network 1, administration program 13 operated in the
14 administration computer 12 which administrates the video
15 program having been divided and stored in magnetic disk
16 units 31 and 32 and the server computers 2. The
17 administration computer-side network interface carries out

1 such process as TCP/IP protocol in the administration
2 computer 12 and realizes an interface between the
3 administration computer 12 and the network 1, and a large
4 capacity storage 15 such as CD-ROM, which is connected to
5 the computer 12 and the original video program 11 is stored
6 therein.

7 Still referring to Fig. 2 only two magnetic disk units
8 are connected to each server computer. Each of the three
9 server computers 2 is connected to two magnetic disk units,
10 respectively, and also connected to the administration
11 computer 12 and a plurality of client computers 5 which are
12 devices on the video-receiving side, via the network 1.
13 Each magnetic disk unit 31 or 32 is divided into block
14 units per a certain amount. Six video programs, denoted by
15 videos 1~6 are stored in 78 blocks denoted by blocks 0~77.
16 Each video program is stored as if data was striped where
17 data has been divided and distributed over the plurality of

1 the magnetic disk units 31 and 32. Video 1 is sequentially
2 stored in the blocks 0~11, and video 2 is sequentially
3 stored in the blocks 12~26. Videos 3~6 are also stored in
4 the blocks, respectively.

5 Referring to Fig. 4 a distributed network 110 includes
6 a plurality of hosts 111 and a shared communication channel
7 112. Each host is coupled to the shared communication
8 channel 112. Each host 111 may act as both a client and a
9 server and uses the distributed network 110, but not all of
10 the hosts need to act as either a client or a server. The
11 downloading process may be called incasting because it can
12 be construed as a reverse of broadcasting. In
13 broadcasting, a file 120 is transmitted to multiple
14 locations generating multiple copies of the file 120. In
15 contrast, in incasting fragments 121 of multiple copies of
16 the file 120 are gathered together to generate a single
17 copy of the file 120. There is a format for creating and

1 storing multiple copies of the files 120 and a protocol to
2 guarantee fast in the sense that it utilizes the maximum
3 available bandwidth for the task and accurate transfer of
4 the requested content/file 120 to a client in the sense
5 that the content of the copied file 120 is the same as that
6 of the stored one. Incasting would constitute the backbone
7 of the distributed network 110.

8 Incasting addresses a key technological issue of how
9 to provide a high-quality service in terms of both accuracy
10 and speed for transferring a file 120, which a client has
11 requested, to the client on the distributed network 110
12 that support content replication. The same content or file
13 120 can reside in several different servers on the
14 distributed network 110. This could be either because the
15 file 120 was created at only one server and then
16 distributed to several others or because the same content
17 was created or procured independently at different servers.

CONFIDENTIAL

1 Incasting will work even if no individual server has
2 the complete file 120, but as long as the complete file 120
3 is collectively available on the whole distributed network
4 110. There is a unique identification tag for each content
5 or file 120 residing on the network. A list of all
6 accessible content/files 120 is either available from one
7 central server or is maintained in a distributed manner.
8 Several servers may contain a complete or partial lists of
9 the contents. Such a list would contain the identification
10 tags of all the contents. For each content/file 120 it
11 would list all the servers that contain a copy of the file
12 120.

13 Referring to Fig. 5 the file 120 is divided into a
14 number of segments 121. Each segment 121 has a secure hash
15 function. The secure hash function is used to compute a
16 message digest, which is then signed. The number of
17 segments 121, their locations, the hash function(s) and the

1 public key(s) for the digital signatures are recorded as
2 attributes of the file 120.

3 The incasting process will work for any existing
4 format for storing files 120 which follows the convention
5 of being byte aligned. Hence, any server can handle a
6 request, where it is asked to transmit blocks of bytes
7 along with start and end indices. For example, a typical
8 request could be for the transmission of M bytes of a file
9 120 starting at the kth byte. However, for guaranteeing
10 the integrity of the files 120 and for avoiding expensive
11 retransmissions of potentially erroneous downloads, the
12 following format for storing files 120 and partitioning the
13 file 120 into a specified number of segments 121 is
14 recommended. For each segment 121, compute a message
15 digest of the contents using a secure hash function. The
16 message digest basically acts as a unique identifier for
17 the contents of the segment 121 and on reception, can be

1 used to guarantee the integrity of the contents of the
2 segment 121. In order to guarantee authenticity (e.g., the
3 fact that the file 120 was indeed created by the owner),
4 one can in addition sign the digest. Thus, if one has the
5 segment 121, the message digest and the digital signature
6 of the file 120, then one can verify authenticity (check
7 that the signature matches the digest) and then check for
8 integrity (i.e., the digest matches the contents of the
9 segment 21). For example, the Secure Hash Standard (SHS)
10 can be used to generate 160-bit message digests for the
11 segments 121. The Digital Signature Standard (DSS) can
12 then be used to generate a 320-bit digital signature of the
13 digest. Other standard hash functions (e.g., MD4 and MD5)
14 and digital signature schemes (e.g., those based on RSA)
15 can be used as well. The number of segments 21 and their
16 starting locations can be stored in the file description.
17 Moreover, if the feature of digital signature is used, then

1 the public key(s) of the owner of the file 20 and the hash
2 function used should also be made available in the
3 description of the files 120.

4 Referring to Fig. 6 each entry for a file 120 in a
5 global list 130 contains all the necessary information
6 about the file 120 so that a client can successfully
7 complete an incasting process. The client wishing to
8 download a file 120 goes through the following step of
9 searching the distributed network 110. The client first
10 searches the global list(s) 130 of content/files 120 (to be
11 referred to as the network directory from hereon) to
12 determine the availability of the desired file 120 on the
13 distributed network 110. It is not necessary that a global
14 network directory be maintained at one or several servers.
15 The network directory could itself be maintained in a
16 distributed fashion (e.g., the scheme adopted in the
17 Gnutella network) in which case, a distributed search for

1 the desired content/file 120 will be carried out. In both
2 cases, the following information is returned to the client.
3 A list of (IP) addresses for the servers where the file 120
4 is located partially or in full. If a server has only
5 parts of the desired file 120, then a succinct description
6 (e.g., start and end byte numbers of contiguous portions of
7 the file 120) of the content stored in the server is also
8 included. If the file 120 is divided into segments 21
9 along with corresponding digest and digital signature, then
10 the client will also receive descriptions of the segments
11 21, and the types of hash functions and public key(s) used
12 for the digital signature. The client now has all the
13 storage information about the desired file 20, but does not
14 know the exact availability of bandwidth at the eligible
15 servers for any download request. Using an adaptive
16 incasting algorithm the client is able to virtually
17 segments the file 120 into a number of distinct parts and

1 requests each part from a distinct server. The exact
2 nature of the virtual segmentation procedure will depend on
3 a number of factors, including, the bandwidth available to
4 the client, any prior knowledge about the bandwidth
5 available to different servers and also the storage format
6 of the requested file 120. Since, these are all very
7 implementation-dependent, specific details of the virtual
8 segmentation procedure are not provided. Different servers
9 will respond at different time intervals to the above-
10 mentioned requests. For example, the servers that have
11 high available bandwidth will respond faster than those
12 with slower access, and some servers might not respond at
13 all. The client can then have an online estimate of the
14 traffic and can change the frequency and size of the
15 requests adaptively. Some servers that do not respond
16 during a pre-specified time interval could be dropped from the
17 list altogether or could be tried again after an interval

1 of time, if the other active servers are not fast enough.
2 This scheme allows complete flexibility and can be used to
3 saturate the available bandwidth of the client. As the
4 above-mentioned adaptive protocol is carried out, the
5 desired file 20 is received in contiguous chunks of bytes.
6 Since the segmentation format of the file 120 is known to
7 the client, it can always check whether any complete
8 segment 21 of the file 120 has been downloaded or not.
9 Once a full segment 121 of the file 120 is downloaded, it
10 can first verify authenticity of the message digest using
11 the digital signature and the public key and then verify
12 the accuracy/integrity of the segment 121 by comparing the
13 downloaded message digest with a digest that it computes on
14 the content of the segment 121 (using a pre-specified hash
15 function). If any of these verification procedures fails,
16 then it discards the whole segment 121 and starts the
17 requests for the bytes in that segment 121 again. Clearly,

1 there is a tradeoff here between the number of original
2 segments 121 in the file 120 and the number of bytes that
3 might be downloaded multiple times. If there are more
4 segments 121 in the file 120, then first the chance that a
5 segment 121 is corrupted is small, and second even if some
6 bytes are corrupted then only a small number of bytes will
7 need to be downloaded again. However, more segments 121
8 would mean a larger overhead in terms of the total size of
9 the file 120. For example, if the Digital Signature
10 Standard is used, then each segment 121 has to have at
11 least an additional 60 bytes: 160 bits (20 bytes) for the
12 message digest and 320 bits (40 bytes) for the digital
13 signature.

14 Incasting allows a client to efficiently download a
15 file 120 from the distributed network 110 by putting
16 together fragments of the file 120 obtained from different
17 servers that maintain partial or complete copies of the

1 desired file 120. While the well-known broadcasting
2 procedure creates copies of the same file 120 at many
3 different destination servers incasting recreates a copy of
4 the file 120 by optimally piecing together fragments of the
5 file 120 obtained from multiple target servers. Incasting
6 provides both a suitable format for storing the files 120
7 and a protocol for gathering the distributed content to
8 create an accurate copy. The same content/file 120 can
9 reside in several different servers on the distributed
10 network 110. This could be either because, the file 120
11 was created at only one server, and then distributed to
12 several others, or because the same content was created or
13 procured independently at different servers. In fact, our
14 invention will work even if no individual server has the
15 complete file 120, but as long as the complete file 120 is
16 collectively available on the whole distributed network
17 110. There is a unique identification tag for each content

1 or file 120 residing on the network. A list of all
2 accessible content/files 120 is either available from one
3 central server, or is maintained in a distributed manner
4 (i.e., several servers contain the complete or partial
5 lists of the contents). Such a list would contain the
6 identification tags of all the contents, and for each
7 content/file 120 it would list all the servers that contain
8 a copy of the file 120.

9 The most frequent use of the distributed network 10 is
10 for downloading purposes. A client looks up the content
11 list, and wants to download a particular content/file 20
12 from the distributed network 10. The existing protocols
13 for this process are extremely simple, and can be described
14 in general as follows. The client or a central server
15 searches the list of servers that contain the desired file
16 20 and picks one such server (either randomly or according
17 to some priority list maintained by the central server) and

1 establishes a direct connection between the client
2 requesting the down load and the chosen server. This
3 connection is maintained until the entire file 20 has been
4 transferred. The exact implementation might vary from one
5 protocol to another; however, the fact that only one server
6 is picked for the transfer of the entire requested file 120
7 remains invariant.

8 The distributed network includes a plurality of hosts
9 and a shared communication channel. Each host has a
10 storage device. U. S. Patent No. 5,630,007 teaches a
11 distributed network which includes a plurality of servers
12 with storage devices and a plurality of clients. In U. S.
13 Patent No. 5,630,007 the servers are distinct from the
14 clients. In this invention the clients and the servers are
15 interchangeable. Each host may act as either a client or a
16 server. A file is divided into a plurality of segments.
17 Each segment is transmitted to the storage devices of

1 several of the hosts and stored in the storage device of
2 the host. Each host is coupled to the shared communication
3 channel. A host acting as a client requests that the other
4 hosts acting as servers and collectively send all of the
5 segments to the requesting client so that the requesting
6 client can gather the segments together in order for the
7 segments to self-assemble and generate a single copy of the
8 file. At least one host has a global list with entries.
9 Each entry contains all the necessary information about the
10 file.

11 From the foregoing it can be seen that incasting for
12 downloading files 120 on distributed networks 110 has been
13 described.

14 Accordingly it is intended that the foregoing
15 disclosure and drawings shall be considered only as an
16 illustration of the principle of the present invention.